

Interpretable tensor-neural networks as topological invariants

D. O. Oriekhov¹,* Stan Bergkamp², Guliuxin Jin², Juan Daniel Torres Luna², Badr Zouggari², Sibren van der Meer, Naoual El Yazidi, and Eliska Greplova²†*QuTech and Kavli Institute of Nanoscience, Delft University of Technology, 2628 CJ Delft, the Netherlands*

(Received 4 December 2025; accepted 10 July 2026; published 20 August 2026)

Much attention has been devoted to the use of machine learning to approximate physical concepts. Yet, due to challenges in interpretability of machine learning techniques, the question of what physics machine learning models are able to learn remains open. Here, we bridge the concept of a physical quantity and its machine learning approximation in the context of the original application of neural networks in physics: topological phase classification. We construct a hybrid tensor-neural network object that exactly expresses the real-space topological invariant and rigorously assess its trainability and generalization. Specifically, we benchmark the accuracy and trainability of a tensor-neural network to multiple types of neural networks, thus exemplifying the differences in trainability and representational power. Our work highlights the challenges in learning topological invariants and constitutes a stepping stone toward more accurate and better generalizable machine learning representations in condensed matter physics.

DOI: [10.1103/grxl-jcbs](https://doi.org/10.1103/grxl-jcbs)

I. INTRODUCTION

Phase classification has become a prototypical benchmark for data-driven analysis of condensed matter physics. The type and complexity of the phase transition dictate the level of complexity of the algorithm one has to employ. This topic has been broadly explored, offering a menu of both supervised and unsupervised techniques, ranging from simple clustering [1–3] to more complex machine learning methods [4–7]. The phase classification problem is most commonly posed like so: we allow our model to view a dataset that is both relevant and straightforwardly obtainable in the scenario we wish to study. We introduce this dataset to a model that has no prior knowledge of underlying physics. Then, we proceed to optimize the model to determine which of the available phases of matter a given sample in the dataset corresponds to.

Within data-driven phase classification, the topological phases are of particular interest [6,8]. The presence of topology in the condensed matter systems is a field of an active research interest [9–18]. In particular, the formulation of appropriate topological invariants is a highly nontrivial complex venture. When it comes to experimental verification [19–27] of the theoretical predictions, there is an additional hurdle: the majority of topological invariants are formulated in the k -space [5,28–30] and their real-space counterparts do not necessarily produce the same phase diagram in a finite size setting [9,31].

Approaching the classification of topological phases from the data-driven perspective, we arrive at a seemingly contradictory observation: topological invariants are inherently global features, but even simple machine learning can immediately distinguish states with and without topological constraints in a both supervised [4] and unsupervised [32] way. Machine learning models have been shown to predict Chern numbers [33–35], winding numbers [6,36,37], and classify the number of topological phases [32,38–40].

We conclude that, on the one hand, it is an exceedingly difficult venture to identify the presence of topology by traditional methods that rely on formulation and evaluation of a global invariant. On the other hand, even a simple neural network can evidently learn to predict the values of these invariants, even from limited data. Are the machine learning models somehow so powerful as to completely bypass the complexity of topological invariants and reliably approximate it from a labeled dataset?

In this work, we set out to rigorously resolve this contradiction and bridge the condensed matter approach of constructing the global invariants with the data-driven machine learning approaches. We do so in three steps: *Step 1*: We construct a hybrid tensor-neural network that *exactly* expresses a known topological invariant: real-space winding number (RSWN) for a general AIII symmetry class system. *Step 2*: We benchmark the performance and generalization of this model for the (extended) Su-Schrieffer-Heeger (SSH) model against different variations of physics-agnostic neural networks. *Step 3*: We analyze the behavior of weights during the training and relate it to the original topological invariant formula.

We show that it is indeed possible to construct a neural network model that learns the topological invariant *exactly* and not simply a proxy within the dataset that allows for effective minimization of the loss function but has no physics meaning. We find that our exact topological neural network approximates the topological invariant with four orders of

*Contact author: d.oriekhov@tudelft.nl†Contact author: e.greplova@tudelft.nl

magnitude better precision than a traditional feedforward neural networks, when trained. However, this achievement comes with a price: by expressing the topological invariant exactly, we leave the overparametrized regime in which neural networks thrive, and the model becomes increasingly sensitive to initialization. Below, we build a side-by-side comparison of designing and training the exact topological model and its physics-agnostic approximation, and specifically highlight the connection between the real-space winding number formula and a neural network.

II. RESULTS

The topological invariant we choose to approximate is the real-space winding number, originally derived for the general AIII class of topological insulators in Ref. [9]. For a one-dimensional system with the chiral symmetry operator Γ , the RSWN reads

$$W = -\frac{1}{L} \text{Tr} \{P_B Q P_A [X, P_A Q P_B]\}. \quad (1)$$

It is expressed through the commutator of Q operator, and projectors $P_{A,B}$ on two sets of sublattices A, B conjugated by the $\Gamma = P_A - P_B$ operator, with the coordinate operator X . The operator $Q = P_+ - P_-$ is composed of projectors on empty and filled bands, and the system length L equals the number of unit cells N times the lattice constant. In this work, we focus on the winding number for half filling; thus, the operators $P_{\pm}$ correspond to positive/negative energy bands. Substituting the components of eigenstates, one finds that formula (1) translates into a polynomial of the fourth order (see Appendix A and Fig. 5).

The prototypical example of the AIII topological insulator is the Su-Schrieffer-Heeger family of Hamiltonians [11,41,42]. These models have two sublattices, exhibit time-reversal symmetry, and their Hamiltonian has the form

$$H = \sum_{n=1}^N (v c_{n,A}^\dagger c_{n,B} + w c_{n,B}^\dagger c_{n+1,A} + J_3 c_{n,B}^\dagger c_{n+2,A} + J'_3 c_{n,A}^\dagger c_{n+1,B}) + \text{H.c.}, \quad (2)$$

where $c_{n,\alpha}^\dagger (c_{n,\alpha})$ denotes the ladder operator of a quasiparticle at lattice site (n, α) , with the unit cell index $n \in [1, N]$ and the sublattice index $\alpha \in A, B$. The ratio of intracell to intercell hopping, v/w , controls the topological phase transition for SSH model. The hopping amplitudes J_3 and J'_3 correspond to the next-nearest-neighbor hopping mechanism, and in their presence, the model is referred to as an extended SSH (ESSH) model. The phase diagram of the SSH model with an even number of sites contains two phases with winding numbers 1 and 0. These phases correspond to $w < v$ and $w > v$ relations of hopping parameters in the infinite chain limit. For the finite-size chain, the hopping ratio corresponding to the phase transition point becomes size dependent. The phase diagram of the extended SSH model contains winding numbers from -1 to 2 [11,31,41–43].

For the family of Hamiltonians described by Eq. (2), the RSWN formula in Eq. (1) can be reduced to the following

expression:

$$W = 2 \sum_{n \notin \{\text{edge st.}\}} \sum_x x ([\Psi_{x,A}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2) + \begin{cases} 2 \sum_{n \in \{\text{hybr. edge states}\}} \sum_x x ([\Psi_{x,A}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2), \\ 4 \sum_{n \in \{\text{loc. edge st., B}\}} \sum_x x [\Psi_{x,A}^{n,+}]^2 \\ -4 \sum_{n \in \{\text{loc. edge st., A}\}} \sum_x x [\Psi_{x,B}^{n,+}]^2. \end{cases} \quad (3)$$

Here, the summation goes over positive energy band eigenstate indices n with energy E_n and coordinates of the unit cells x . The $\Psi_{x,A/B}^{n,+}$ is a component of the eigenstate n with $E_n > 0$ at the unit cell with coordinate x and site A/B . The summation includes differences of wavefunctions on the different sublattices A and B within the same unit cells. The derivation of this formula is shown in detail in Appendix A. Because of the finite-size effect on edge-state localization properties [31], the formula contains two types of contributions. First type corresponds to hybridized edge states, and second to the single-side localized states, distinguished by the corresponding edge of chain and sublattice family of the main wavefunction weight A/B .

Notable aspect of expression in Eq. (3), that even though it is a rather complicated summation, it is a linear combination of the products of the squared SSH wavefunction amplitudes with coordinates. Equation (3) can thus immediately inform our neural network construction for RSWN. Our architecture is shown in Fig. 1(a). We first preprocess the wavefunction amplitudes via a tensor layer that forms the two-term product pairs. These are then connected, via a single layer of weights, to the output neuron that stores the value of RSWN, W . From now on, we refer to this model as *topological*, since for specific fixed values of its weights [blue lines in Fig. 1(a)], the formula (3) is expressed exactly. In Fig. 1(b), we show a natural generalization of the topological model: we maintain the tensor layer preprocessing and connect it to a two-layer generic neural network with rectified linear unit (ReLU) activation functions. This network no longer expresses RSWN exactly, but takes correctly paired wavefunction amplitudes as its inputs and has a larger space of trainable parameters to approximate RSWN. We refer to the network as *preprocessed*. Finally, in Fig. 1(c), we show a fully connected feedforward neural network (FNN) that takes the wavefunction amplitudes as an input individually and is trained to approximate RSWN (for architecture details, see Appendix E).

In Fig. 2, we show the loss function, mean-squared error (MSE) with respect to the correct value of RSWN, as a function of training epoch. We train on the dataset of approximately 140k samples. Each sample is a two-dimensional array of wavefunction components ordered as a function of lattice site and eigenstate index. Each sample is generated from a Hamiltonian in Eq. (2) with randomly chosen values of hopping amplitudes. The generated set of Hamiltonians has approximately equal distributions over different winding number classes. Half of the dataset is generated using the SSH Hamiltonian, and the other half is generated using the ESSH Hamiltonian. The full dataset is divided into training and validation datasets comprising of 80% and 20% of the data, respectively. In addition, we discard the samples

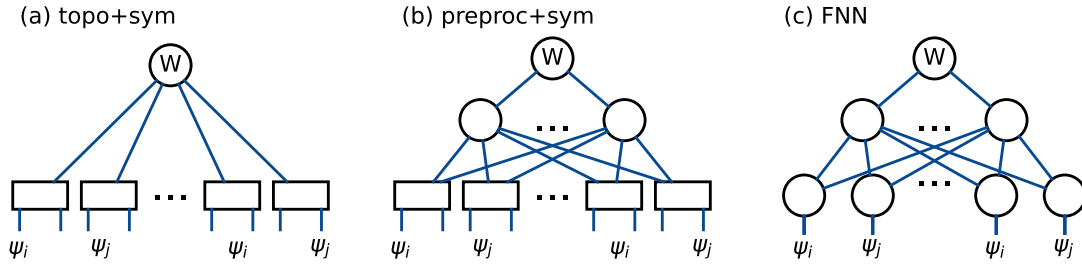


FIG. 1. Neural network architectures: (a) *Topological* neural network that combines a tensor layer and one neural layer that exactly reproduces the structure of real-space winding number for SSH-like two-band models within the AIII symmetry class. (b) *Preprocessed* neural network: the tensor layer remains included, but the remainder of the network consists of multilayer classifier that no longer exactly corresponds to the topological invariant. (c) *Feedforward* neural network is a fully connected classifier without the preprocessing layer and without the default correspondence to the topological invariant. Input size for (a) and (b) networks scales as N^2 for N unit cells. The input size for network (c) is $4N^2$.

(ca. 7000) that lie too close to RSWN phase transition and are prone to numerical instabilities during RSWN evaluation (see Ref. [31] for details).

We show the loss of the topological model in blue, the preprocessed model in red [trained with stochastic gradient descent (SGD)] and in orange (trained with Adam optimizer), and FNN in dark green (SGD) and light green (Adam). Panel (a) shows training, and panel (b) shows validation. We see that the topological model performs by far the best and shows the fastest convergence. The initialization of the topological model was performed in a deterministic way: first, we take the set of weights that match exactly the RSWN formula in Eq. (3), where $w_i = 1$ for all i . Second, we randomize the weights by adding a random tensor generated from a Gaussian distribution (with mean 0.4 and variance 0.15) with a fixed seed for reproducibility. Then, we begin to train.

In Fig. 2, we see that the preprocessed networks show relatively the best generalization. While the topological network has by far the lowest loss on the training set, the validation loss flattens to a comparatively high number relatively quickly. For the preprocessed network, the training and validation behavior remains very similar. The feedforward network shows the

worst loss of all models on both the training and validation sets. At this point, we can conclude that while the topological network reaches impressively low loss on the training set, it might not have the best generalization properties. The preprocessed network, on the other hand, appears to be the most flexible. We will return to this point later during the further analysis of these networks. The next critical aspect we need to investigate is the generalization of our model. In principle, it is simple to construct a model that has a very low loss on a given dataset. The challenge is to make the model generalize correctly. This aspect is particularly important when one wishes to use data-driven techniques to reliably determine topology: a key feature of topological invariants is that they robustly predict topological phase even in the presence of disorder. In Appendix B, we explain in depth a simple example of a poor generalization: we show that most of the amplitudes of any eigenstate of SSH models are monotonic functions of the ratio v/w that determines the phase transition. As such, an example of a model that can predict whether a given state is in the trivial or topological phase is a value of almost any single wavefunction amplitude. We show that, unsurprisingly, such a model is not resilient against disorder and becomes invalid almost immediately. The in-depth study of generalization of physics-agnostic neural networks on the SSH Hamiltonian was shown in Ref. [44].

In Fig. 3, we show an analysis of the generalization for all three types of models in Fig. 1, between SSH and ESSH models and in the presence of disorder. The red color family denotes SSH and blue color family denotes ESSH. The dashed lines show the values of FNN neural network. In Fig. 3(a), we show MSE loss as a function of the initialization variance for the topological model trained on SSH half of our dataset. Here, we follow the initialization strategy explained above, but vary the mean of the Gaussian distribution from which we sample the weights from. Unsurprisingly, SSH at zero disorder ($d = \delta t/t = 0$) exhibits very low MSE values across the initialization range with slight increase as a function of the initialization variance. This increase is upper bounded by the MSE value of the physics-agnostic FNN trained on this dataset. Disordered data for SSH and both disordered and nondisordered data for ESSH behave similarly, albeit with bigger absolute value, and are upper bounded by their respective FNN. The relative disorder d of hopping parameters v, w in SSH model and J_3, J'_3 in ESSH model was equal

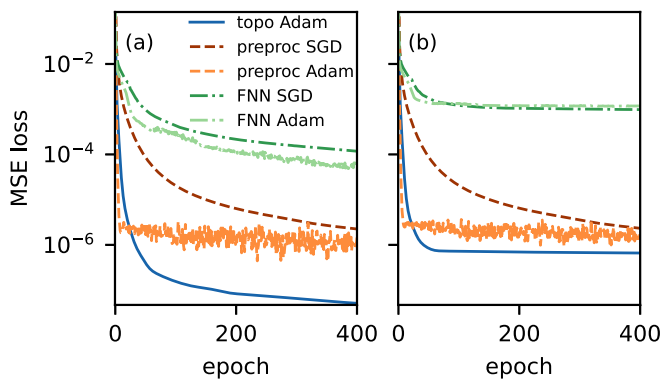


FIG. 2. MSE loss for the RSWN regression: Panel (a) shows MSE loss on the training dataset that comprises 80% of the data and panel (b) on the validation dataset, which consist of 20% of the data. Three models were used to learn the float value of RSWN. The topological model (“topo”) is in blue, and the preprocessed model (“preproc”) is in red and orange for SGD and Adam, respectively. FNN model is in dark (light) green for SGD (Adam).

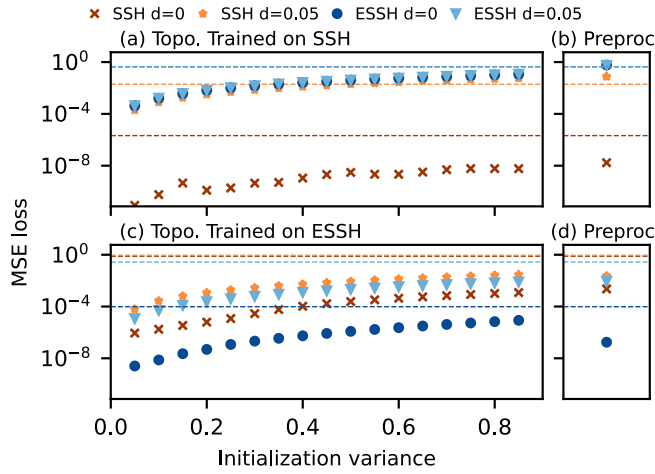


FIG. 3. Generalization of the topological, preprocessed, and FNN models: each neural network is trained on one nondisordered $d = 0$ type of dataset corresponding to a model Hamiltonian of SSH [panels (a) and (b)] or ESSH [panels (c) and (d)], and tested on another Hamiltonian dataset as well as both disordered types (color is indicated in legend). Model types are indicated in the titles of panels. We train a topological model [panels (a) and (c)] for different initialization variances and a preprocessed model [panels (b) and (d)] averaged over five different seeds. The FNN model results are the dashed lines in each panel.

to 0.05 in all data samples. In Figs. 3(b)–3(d), we perform analogous analysis. We train the preprocessed model on SSH in panel (b), topological model on ESSH in panel (c), and the preprocessed model on ESSH in panel (d), respectively. We observe that topological models have better generalization properties, especially at lower initialization variances. Their error slowly increases until it reaches roughly the value of the preprocessed network, which is in turn always upper bounded by values of FNN.

A common way to gain insight into the interpretability of a machine learning model is to analyze its weight space [45–47]. In the instance of the topological neural network and its corresponding formula in Eq. (3), the weight space analysis

can be particularly interpretable. First, the architecture itself is simple and after tensor preprocessing, it only consists of a single trainable layer. Specifically, the kernel of the tensor is multiplied with the weights of the single layer, $\Delta_{n,i}$, and a bias is added. This hybrid tensor-neural network then outputs the topological invariant, W . More importantly, the symmetry properties of Eq. (3) dictate that all the weights need to lie on the line $\Delta_{n,x} + \Delta_{n,-x} = 2$, where index n corresponds to the eigenstate and index x to the site of the chain (see Appendix A for details). In Fig. 4, we visualize the weight trajectories during the training for three different initializations [panels (a), (b), and (c), respectively]. The orange dots specify the initialization point for each weight, the blue dots the final point during training, and the light blue line the trajectory. Each panel corresponds to the different instance of Gaussian initialization initialized with a mean of 1.0, 1.5, and 0.5, respectively. All distributions have a variance of 0.3. For this analysis, we added disorder to our initial dataset. Specifically, we introduce an inversion symmetric disorder to the parameters w, v, J_3, J'_3 . Disorder strength, $d = \delta\{w, v, J_3, J'_3\}/\{w, v, J_3, J'_3\}$ was chosen to be $d = 0.05$ for 75% of the samples and $d = 0.08$ for 25% of the samples. We constructed this disorder explicitly to violate symmetries in the dataset, such that we can visualize the weights converging on a simple line instead of a more complicated pattern. Additionally, we combined two disorder patterns to prevent the model from fixating on specific amplitudes within the dataset with a uniform disorder (see Appendixes B and C). Here, we used Adam optimizer with learning rate 5×10^{-4} for the first 20 epochs and 10^{-4} for the rest of the training. This selection was made following to the comparison shown in Fig. 2. See also Appendix E for further details.

In Fig. 4(a), we see that the weights converge toward the required expression, while in panels (b) and (c) only a portion of the weights converges on the correct line, while the rest of the weights converge to another linelike structure, albeit with the constant offset to the actual symmetry line. This means that the network converged to a suboptimal local minimum. It found an approximate yet not precise RSWN formula with a similar structure in the process. We attribute this

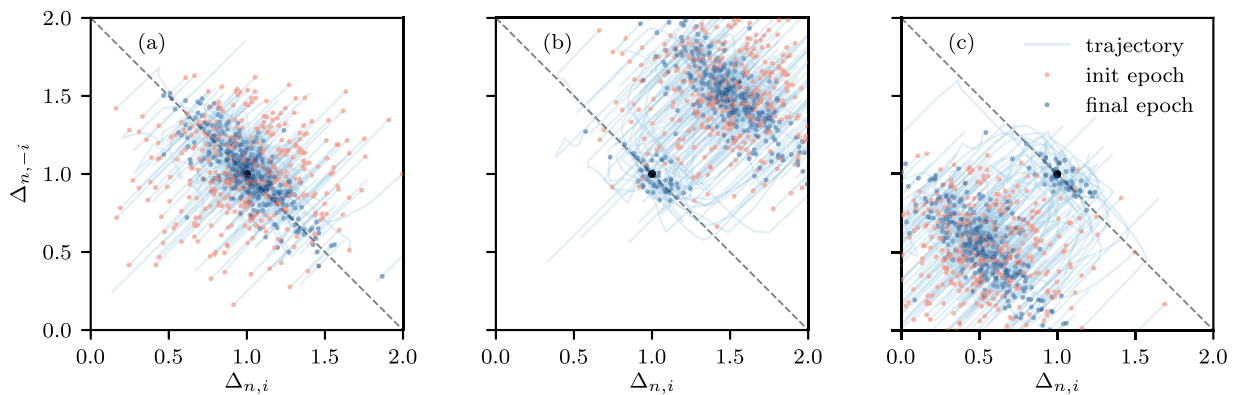


FIG. 4. An evolution of the topological model weights during training: dots indicate pairs of weights $(\Delta_{n,i}, \Delta_{n,-i})$ linked via inversion symmetry; the orange dots are initial values of each weight, the blue line corresponds to the training trajectory, and dark blue dot is the final point. (a) the model initialization is centered around the correct RSWN formula, and the weights are initialized from the Gaussian distribution centered around 1.0; (b) the Gaussian distribution for weights initialization is shifted away from the correct RSWN formula and is centered around 1.5; (c) the Gaussian distribution for the weights initialization is centered around 0.5. The training is done for chain size $N = 28$.

behavior to the stochastic nature of the optimizer, the size of the model, and generally very small values of the loss. Specifically, our loss function is reaching extremely small values making different local minima practically indistinguishable. Additionally, compact size of the model means that it might be challenging to reach all the points in the parameter space. As such, the topological networks are prone to converge to relatively precise but slightly incorrect RSWN formulas.

III. CONCLUSIONS AND OUTLOOK

In this work, we set out to answer the question “Why is topology hard to learn?” The reason for this difficulty is that trained machine learning models most commonly learn a dataset-dependent proxy rather than the topological invariant itself. Here, we have shown how to construct a trainable object that learns the topological quantity exactly. However, constructing such precise topological invariant representation comes hand in hand with hindrances in trainability. While the topological network we construct is very sensitive to optimizer selection and initialization, it exemplifies the difference between learning the physical quantity itself and learning its machine learning approximations. Interestingly, even when we optimize all the hyperparameter conditions for the topological network to converge to the correct formula, an initialization too far away from this optimum makes it converge to an incorrect formula. This is not surprising, as many minima result in very low cost and the neural network of course does not have a way to distinguish between all these models. Formulating the precise distinction between the three benchmark models, we introduced here, feedforward, preprocessed, and topological, opens up an avenue to the more informed and interpretable application of machine learning models in condensed matter physics.

Going forward, we could apply the approach presented here to enhance interpretability for the systems with unknown topological order parameters. One could combine the tensor network layer technique with the multikernel approach from convolutional neural networks. In particular, combining several tensor network layers of different rank and performing bottleneck procedure [48], one could extract the main structure of topological invariant out of several candidates. There is also exciting possible connection with Kolmogorov-Arnold networks (KANs) [49]. The key ingredient of the topological networks presented in this work is the tensor transformation layer. This layer converts a nonlinear polynomial structure of topological invariant into a linear task, well suited for linear perceptron neural nets. Kolmogorov-Arnold networks have been shown to be well suited for approximating polynomial functions. In this regard, we point out that tensor-neural construction of our topological network could be viewed as a bridge between KANs and linear perceptron. Namely, a first tensor layer could be composed out of several KAN layers described in Ref. [49].

ACKNOWLEDGMENTS

We are thankful for fruitful discussions with Ana Silva, Thomas Spriggs, Anton Akhmerov, Mohammed Boky, Marco Marciani, and Anna Dawid. This work is part of the project

Engineered Topological Quantum Networks (Project No. VI.Veni.212.278) of the research program NWO Talent Programme Veni Science domain 2021, which was financed by the Dutch Research Council (NWO). D.O.O. acknowledges the support by the Kavli Foundation. G.J. acknowledges the research program “Materials for the Quantum Age” (QuMat) for financial support.

DATA AVAILABILITY

The code and generated data for all simulations performed in the paper can be found in Ref. [50].

APPENDIX A: DERIVATION OF REAL-SPACE WINDING NUMBER FOR THE FAMILY OF SSH MODELS

In this Appendix, we demonstrate how the RSWN formula derived for the general AIII class topological insulator in Ref. [9] could be reduced to a unique form for the SSH-type family of insulators. The derivation includes the following steps: (1) definition of eigenbasis of chiral symmetry operator for SSH-type models with two sublattices; (2) transformation of general RSWN expression to this basis by using cyclic properties of trace; (3) derivation of restricting identities that follow from orthogonality condition of eigenstates with different energies; (4) application of these identities to downfold the power of polynomials in RSWN formula from fourth order to second order; and (v) presentation of additional restrictions from inversion symmetry on the lattice.

The RSWN for general AIII class topological insulator reads

$$W = -\frac{1}{L} \text{Tr} \{P_B Q P_A [X, P_A Q P_B]\}. \quad (\text{A1})$$

It is expressed through the commutator of Q operator, and projectors $P_{A,B}$ on two sets of sublattices A, B conjugated by the $\Gamma = P_A - P_B$ operator, with the coordinate operator X . The operator $Q = P_+ - P_-$ is composed of projectors on empty and filled bands, and the system length L equals to number of unit cells N times the lattice constant.

Due to the chiral symmetry Γ that acts in sublattice space, the eigenstates have the following form:

$$\begin{aligned} \Psi^{n,+} &= \sum_x (\Psi_{x,A}^{n,+} |x, A\rangle + \Psi_{x,B}^{n,+} |x, B\rangle), \\ \Psi^{n,-} &= \sum_x (\Psi_{x,A}^{n,+} |x, A\rangle - \Psi_{x,B}^{n,+} |x, B\rangle). \end{aligned} \quad (\text{A2})$$

Here, we took into account that Hamiltonian has a block-antidiagonal form and $\Gamma \Psi^{n,+} = \Psi^{n,-}$, with $\Gamma = \sigma_3 = \text{diag}(1, -1)$.

The operators $Q_{AB/BA}$ have the form after substitution of positive and negative energy projectors, and cancellation of equal terms:

$$Q_{AB} = 2 \sum_x \sum_{x'} \sum_n |x, A\rangle \Psi_{x,A}^{n,+} \Psi_{x',B}^{n,+} \langle x', B|, \quad (\text{A3})$$

$$Q_{BA} = 2 \sum_x \sum_{x'} \sum_n |x, B\rangle \Psi_{x,B}^{n,+} \Psi_{x',A}^{n,+} \langle x', A|. \quad (\text{A4})$$

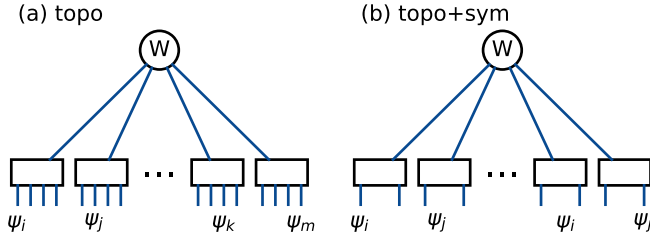


FIG. 5. A change of architecture of a “topology-inspired” neural network under the properties of SSH-type modes at half filling. The property of bands under chiral symmetry is included to change the tensor network layer from a four-leg type to a two-leg type. The corresponding sizes of input and the number of weights in the regression layer are (a) N^4 and (b) N^2 .

Next, the winding number reads

$$\begin{aligned}
 W &= \frac{1}{L} \text{Tr}[Q_{BA}(\hat{X}Q_{AB} - Q_{AB}\hat{X})] \\
 &= \frac{1}{L} \text{Tr}[Q_{BA}\hat{X}Q_{AB}] - \text{Tr}[Q_{AB}\hat{X}Q_{BA}] \quad (\text{A5}) \\
 &= \frac{4}{L} \text{Tr} \left[\sum_x \sum_{x'} \sum_n |x, B\rangle \Psi_{x,B}^{n,+} \Psi_{x',A}^{n,+} \langle x', A|x' \right. \\
 &\quad \times \sum_{x''} \sum_{n'} |x', A\rangle \Psi_{x',A}^{n',+} \Psi_{x'',B}^{n',+} \langle x'', B| \left. \right] \\
 &\quad - \frac{4}{L} \text{Tr} \left[\sum_x \sum_{x'} \sum_n |x, A\rangle \Psi_{x,A}^{n,+} \Psi_{x',B}^{n,+} \langle x', B|x' \right. \\
 &\quad \times \sum_{x''} \sum_{n'} |x', B\rangle \Psi_{x',B}^{n',+} \Psi_{x'',A}^{n',+} \langle x'', A| \left. \right]. \quad (\text{A6})
 \end{aligned}$$

$$\sum_x \Psi_{x,A}^{n,+} \Psi_{x,A}^{n',+} = \sum_x \Psi_{x,B}^{n,+} \Psi_{x,B}^{n',+} = \begin{cases} \frac{1}{2} \delta_{n,n'}, & n, n' \notin \{\text{edge state indices}\}, \\ \delta_{n,n'} \text{ for } A/B, 0 \text{ for } B/A, & n = n' = \text{localized edge state}. \end{cases} \quad (\text{A10})$$

Substitution of these relations into RSWN formula reduces to the following sum of two-term products:

$$W = 2 \sum_{n \notin \{\text{edge st.}\}} \sum_x x ([\Psi_{x,A}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2) + \begin{cases} 2 \sum_{n \in \{\text{hybr. edge states}\}} \sum_x x ([\Psi_{x,A}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2), \\ 4 \sum_{n \in \{\text{loc. edge st., B}\}} \sum_x x [\Psi_{x,A}^{n,+}]^2 - 4 \sum_{n \in \{\text{loc. edge st., A}\}} \sum_x x [\Psi_{x,B}^{n,+}]^2. \end{cases} \quad (\text{A11})$$

Such simplification significantly reduces the amount of redundant weights in “topo” NN, making their total amount to be N^2 . The corresponding structure of the network is presented in Fig. 5(b). However, this formula is still not unique for the SSH chain. Due to inversion symmetry, $\Psi_{x,A}^{n,+} = \Psi_{L-x,B}^{n,+}$, for each n except the localized edge states. And, the localized edge states themselves also transform into each other under a symmetry operation. For example, the first part of the formula can be replaced by the following

This formula could be converted into the expression presented in the main text,

$$W = \frac{4}{L} \left(\sum_x \sum_{x'} \sum_n \sum_{n'} \Psi_{x,B}^{n,+} \Psi_{x',A}^{n,+} x' \Psi_{x',A}^{n',+} \Psi_{x,B}^{n',+} - \sum_x \sum_{x'} \sum_n \sum_{n'} \Psi_{x,A}^{n,+} \Psi_{x',B}^{n,+} x' \Psi_{x',B}^{n',+} \Psi_{x,A}^{n',+} \right). \quad (\text{A7})$$

From this expression, the individual four-term products could be taken as inputs into “topo” neural network [see Fig. 5(a)].

Due to only two sublattices, the following identities hold:

$$\begin{aligned}
 \delta_{n,n'} &= \langle \Psi^{n,+} | \Psi^{n',+} \rangle = \sum_x (\Psi_{x,A}^{n,+} \Psi_{x,A}^{n',+} + \Psi_{x,B}^{n,+} \Psi_{x,B}^{n',+}), \\
 0 &= \langle \Psi^{n,+} | \Psi^{n',-} \rangle = \sum_x (\Psi_{x,A}^{n,+} \Psi_{x,A}^{n',+} - \Psi_{x,B}^{n,+} \Psi_{x,B}^{n',+}). \quad (\text{A8})
 \end{aligned}$$

Taking sum and difference, we obtain the following results:

$$\sum_x \Psi_{x,A}^{n,+} \Psi_{x,A}^{n',+} = \sum_x \Psi_{x,B}^{n,+} \Psi_{x,B}^{n',+} = \frac{1}{2} \delta_{n,n'}. \quad (\text{A9})$$

This result does not hold for localized edge states. Instead, such combinations give 1 or 0 (or a random float number in the vicinity of the phase transition). Thus, the corrected expression is

expressions:

$$\begin{aligned}
 W &= 2 \sum_n \sum_x x ([\Psi_{L-x,B}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2) \\
 &= 2 \sum_n \sum_x x ([\Psi_{x,A}^{n,+}]^2 - [\Psi_{L-x,A}^{n,+}]^2) \\
 &= 2 \sum_n \sum_x x ([\Psi_{L-x,B}^{n,+}]^2 - [\Psi_{L-x,A}^{n,+}]^2). \quad (\text{A12})
 \end{aligned}$$

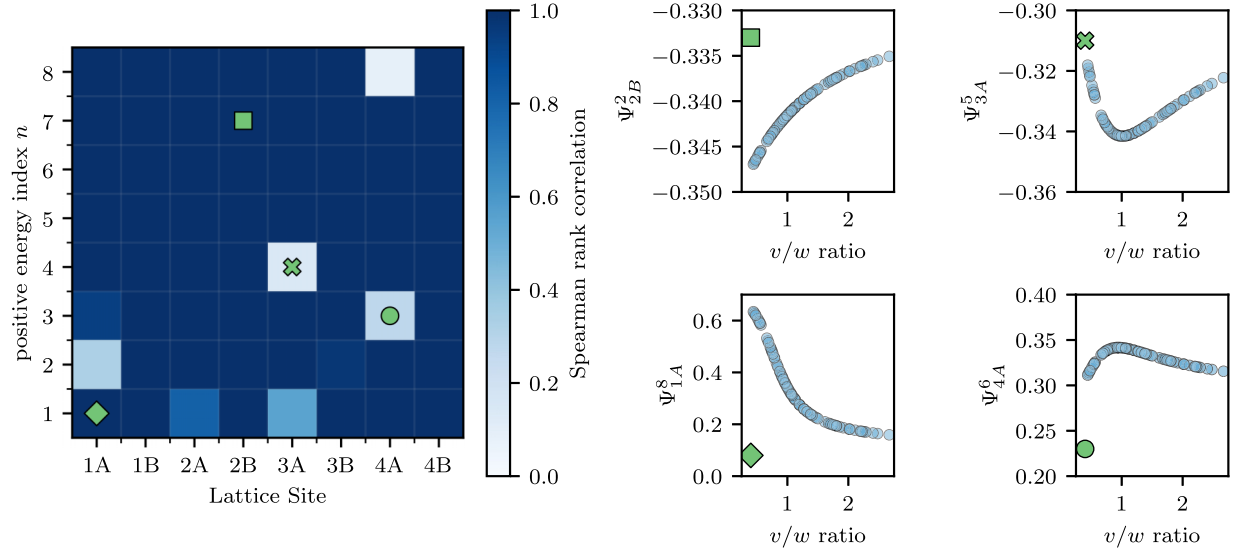


FIG. 6. Left: A heatmap of the rank correlation coefficient at zero disorder is shown for individual components of SSH eigenstates. Axes correspond to lattice site and energy index of eigenstate. The plot is restricted to positive energies and half of chain, as the other indices correspond via particle-hole and inversion symmetries. Right: Four panels, marked by a green square, cross, diamond, and circle, correspond to the similarly marked lattice sites in the left panel. Each panel shows the relationship of the amplitude values and v/w for the respective wavefunction amplitude-lattice site combinations $\psi_{2,2B}$, $\psi_{5,3A}$, $\psi_{8,1A}$, $\psi_{6,4A}$.

Taking sums and differences of these expressions, we find that RSWN formula could be further restricted to

$$W = 4 \sum_{n \notin \text{edge states}}^L \sum_{x=0}^{L/2-1} x([\Psi_{x,A}^{n,+}]^2 - [\Psi_{x,B}^{n,+}]^2). \quad (\text{A13})$$

The corresponding amount of inputs to NN is reduced to $N^2/2$ values.

APPENDIX B: CLASSIFICATION BY CORRELATION OF WAVEFUNCTION AMPLITUDES WITH HOPPING PARAMETER

This Appendix summarizes the results originally obtained in Ref. [51] with the focus on why most neural networks do not automatically learn to capture topological aspects of condensed matter systems.

As discussed in the main text, the FNN models take as input the wavefunction amplitudes and output the winding number or topological class. In this Appendix, the inputs are composed of eigenstates of the SSH Hamiltonian, sorted by their respective energies. The label of the topological class is defined by the RSWN formula. The winding number in both finite and infinite systems is a monotonic function of hopping parameters ratio, v/w [9,31]. Thus, if at least one amplitude is also a monotonic function of the v/w , the classification can be done with a single-perceptron neural network. As an example, neural network that has very high corresponding weight and bias placed exactly at the phase transition point would classify both trivial and topological classes instantly.

To check the occurrence of such amplitudes within an input eigenstate, we perform numerical tests on SSH chain with $N = 8$ unit cells.

1. Rank correlation

For the selected eigenstates we tested (shown in Fig. 6, right panel), we find multiple amplitudes to be a monotonic function of the v/w . This ratio is also the defining factor in the classification of the system as trivial or topological. Thus, any deep machine learning model could simply learn the ranking of the limited amount of amplitude data points and retrieve a threshold value for RSWN value crosses 0.5 to identify the phase.

To test this numerical observation systematically, we determine the Spearman correlation rank of every amplitude of the calculated eigenstates. Positive correlations imply that as v/w increases, so does wavefunction amplitude Ψ^i , and vice versa for negative correlations. If a threshold of the amplitude were used to classify the phase, then this rank correlation would be an upper limit of the accuracy; i.e., a rank correlation of 1 or -1 would mean perfect phase classification.

As can be seen in the left panel of Fig. 6, almost all components have an absolute value of rank correlation equal to 1. In the four panels on the right, the wavefunction amplitude behavior is shown for some specific wavefunction-lattice site combinations ($\Psi_{2,B}^2$, $\Psi_{3,A}^5$, $\Psi_{1,A}^8$, $\Psi_{4,A}^6$). These panels show a smooth function of the amplitudes as a function of v/w . Two panels show perfectly monotonic behavior, while remaining two show nonmonotonic behavior.

This analysis suggests that one of the reasons the FNN models may not learn the underlying topological patterns related to the calculation or estimation of topological information or invariants is that these data are easily classified. Therefore, the models do not need to extract these complex patterns to achieve high classification accuracy. The amplitudes are directly related to the hopping parameters and, therefore, the topological phases themselves. Using more complicated neural network architectures for this specific

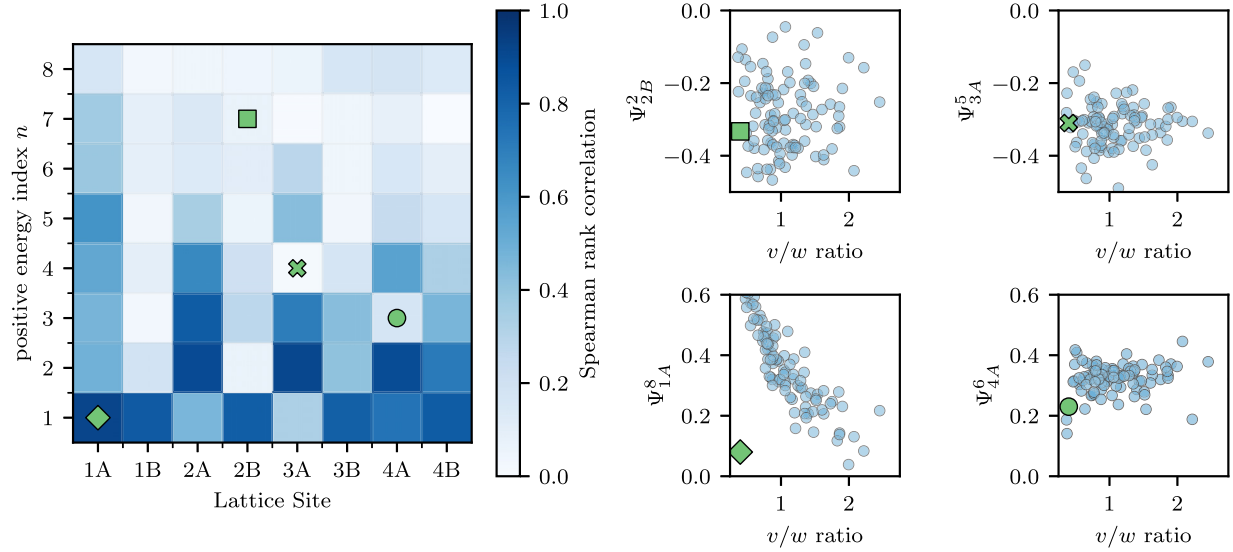


FIG. 7. Left: A heatmap of the rank correlation coefficient for Gaussian disorder with standard deviation 0.1 is shown for individual components of SSH eigenstates. Axes correspond to lattice site and energy index of eigenstate. Right: Four panels marked by a green square, cross, diamond, and circle correspond to the similarly marked lattice sites in the left panel. Each panel shows the relationship of the amplitude values and v/w for the respective wavefunction amplitude-lattice site combinations $\Psi_{2,2B}^2$, $\Psi_{5,3A}^5$, $\Psi_{8,1A}^8$, $\Psi_{6,4A}^6$.

supervised learning task will therefore also yield the same results.

This conclusion motivates us to study the performance of machine learning (ML) models discussed in the main text on disordered datasets as well. Disorder adds a “noise” into eigenstate images and breaks the possibility of classification with a single perceptron. This is discussed in detail below.

2. Disorder in hopping parameters v/w

To look at the effect of disorder of the individual hopping parameters on the amplitude data, the same rank correlation and corresponding amplitude scatter plots are given for a fixed relative value of disorder. The disorder is created by adding Gaussian zero-mean white noise (ZMWN). The relative disorder values indicate the width of the distribution (standard deviation) of the ZMWN that is added to each hopping parameter individually in the real-space Hamiltonian before diagonalization. In other words, all off-diagonal hopping elements in the Hamiltonian get a random noise term added.

When ZMWN with a standard deviation of 0.10 is added, the data become increasingly noisy, with the upper and lower energy states barely being correlated with labels (seen on the left of Fig. 7). In the right panels of Fig. 7, the four selected wavefunction amplitude-lattice site combinations increasingly show less correlation between the amplitudes and v/w . This indicates that the simple correlations between eigenstates and labels are removed from the dataset. Then, the FNN classifier trained on the clean dataset should demonstrate low performance on the disordered dataset—which appears in Figs. 2 and 3.

APPENDIX C: SVD ANALYSIS OF FNN CLASSIFIER

This Appendix contains results obtained during bachelor thesis project of Zougari [52]. In this Appendix, we analyze

the correlations between individual eigenstates of SSH model and the topological phase. Based on these correlations, we construct a very simple feedforward NNs that classify data with more than 90% accuracy. We show that the SVD pruning can be used to build such networks from a standard classifier. The main drawback of such networks is that their performance drops dramatically on disordered systems (not unlike the example we described in the previous Appendix).

The output Y_i of the first layer of neural network is given by the following expression in terms of input X_j , weight matrix Δ_{ij} , and bias B_j :

$$Y_i = \text{ReLU}(\Delta_{ij}X_j + B_i), \quad (\text{C1})$$

The same form holds for the next layers as well. The SVD applied to weight matrix reads

$$\Delta_{ij} = U_{ik}D_{kn}V_{nj}, \quad (\text{C2})$$

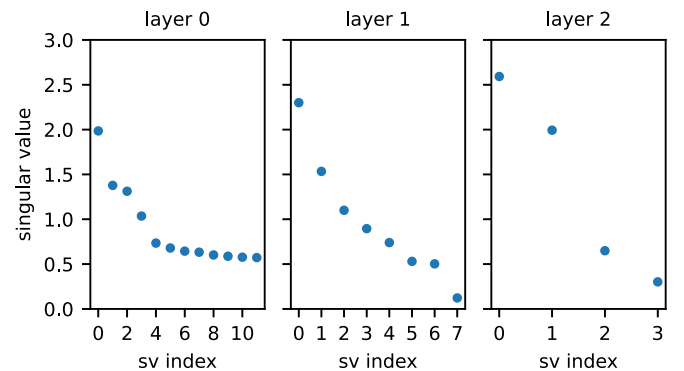


FIG. 8. Singular values of the three layers for an FNN classifier. The architecture of the classifier for the $N = 28$ -unit cells ESSH chain is input $(2N)^2$, first hidden dimension 12, second 8, and output 4. The accuracy after 20 epochs of training was 99.7% for the dataset of size $\sim 50\,000$, with 25% selected for validation.

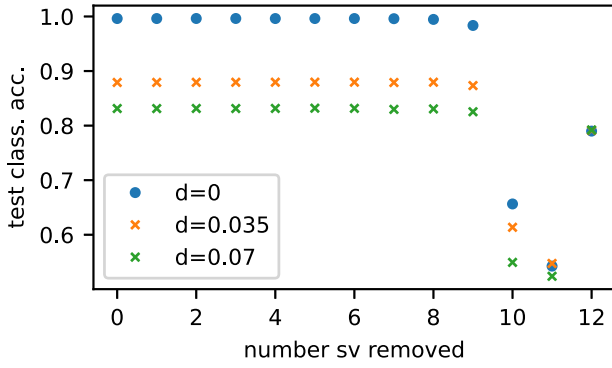


FIG. 9. Accuracy of classification when several of the lowest singular values are replaced by zeros in the first layer weight matrix. The plot demonstrates that only the three to four largest singular values are responsible for the 99% precision of classification. This suggests a possibility of a significant reduction of the total amount of weights present in the network.

where D_{kn} is a rectangular matrix with singular values on nonzero diagonal and both U , V matrices are square matrices. This form of representation for weight matrix Δ suggests that it is possible to extract the linear feature-map transformations from the vectors within V . If we remove a number of small singular values from D , we get a remaining set of corresponding relevant vectors in U , V . They form a projector operator:

$$\tilde{\Delta}_{ij} = \sum_{d \in \{|D| > \text{const}\}} U_{id} D_{dd} V_{dj}. \quad (\text{C3})$$

The singular values within each layer, for an example, classifier with three layers are shown in Fig. 8. The dependence of classification precision on singular values is shown in Fig. 9. The vectors V_{dj} select significant superpositions of initial wavefunctions that play the main role in successful classification. Note that the orthogonality of eigenstates allows for inferring the direct importance of each state in such superposition. For example, it is reasonable to look only at edge nodes for zero-energy states. That is clearly shown in the density plots in Fig. 10, as depicted by red arrows. An additional property of such a projection is that it performs a significant reduction of parameter space size between layers, playing the role of a large convolution kernel applied to the whole image.

Based on Figs. 9 and 10, one concludes that the simplest FNN classifier, which maintains 98% accuracy for ESSH phase classification, requires a hidden dimension of the second layer to be equal 2. The corresponding pruned NN has just $N^2 \times 2 + 2 \times 8 + 8 \times 4$ parameters. This agrees well with the results in Appendix B for the SSH model.

Next, we show an intensity plot of individual “feature map” vectors that correspond to the largest singular values in Fig. 8(a). The intensity plot shows that these vectors mainly check for the presence of localized edge states, as indicated by the arrows. In addition, they compare oscillatory patterns for particular eigenstates that are correlated the most with the topological phase label. This way of classification does not lead to learning topology. However, it still produces

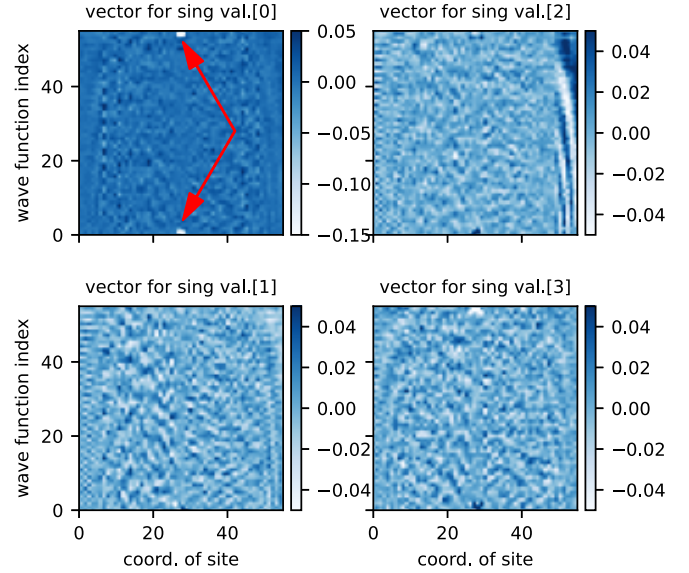


FIG. 10. The four vectors of matrix V in SVD decomposition of the first layer that correspond to the largest singular values. These four vectors are enough to achieve the 99% accuracy of classification. The arrows show that these vectors are working like a feature maps that check for the existence of edge states.

high accuracy as some states (in particular, the lowest energy states, which are transformed to edge states after phase transition) have a qualitative change in their structure. In Appendix B, we show that for SSH model many of the states and their individual components are well correlated with the topological phase label. These components could be used to efficiently classify data with almost single-perceptron-type networks.

APPENDIX D: CLASSIFICATION BENCHMARK

As an additional benchmark, we test the performance of all NNs (topological, preprocessing, and FNN) on the classification task for the ESSH model. Specifically, we will not try to regress on the value of the RSWN; we will simply try to predict in which of the four known classes the input wavefunction lies. For ESSH model, there are in total four classes of winding numbers in finite-size geometry [11,31,41–43]. The dataset contained 100k images of ESSH eigenstates without disorder. Each half of dataset was taken along different line in phase diagram of finite size chain of size $N = 28$. The benchmarks of classification are shown in Fig. 11. The results show the “topo” network to perform much worse comparing to FNN classifier, while “preproc” network demonstrates the best performance. This could be explained by the fact that “topo” network is designated for perfect solution of regression task. To convert it into classifier, additional layer that converts outputs into classes was used. For precise architecture description, see Ref. [50] and Appendix E. The extra amount of parameters in “preproc” network and its multilayer structure allowed us for better learning and prediction of integer labels of classes.

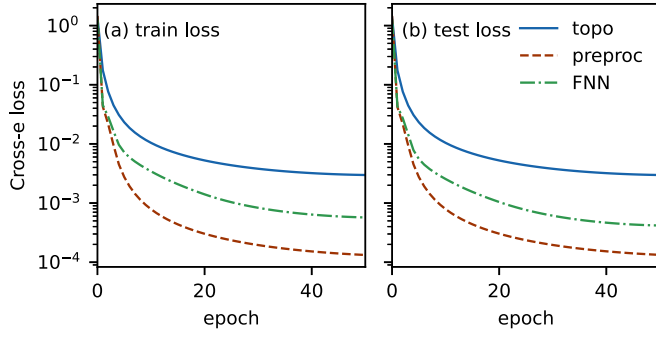


FIG. 11. Benchmarking of three models (topological, preprocessed with middle layer size 12, and FNN with hidden layer sizes 12 and 8) on the one-hot ESSH classification task. Initialization is fully random for both models. Test dataset size 0.15, total dataset size 95k samples. All classes—RSWN 0, 1, 2, -1 —are present in equivalent proportions (e.g., same scale).

APPENDIX E: NEURAL NETWORK ARCHITECTURES AND OPTIMIZER PERFORMANCE

In this Appendix, we provide a technical description of the neural networks used in this study and the corresponding optimizers.

The loss functions that are optimized during training are MSE loss for regression task and Cross-entropy loss for classification task, both taken from PyTorch. Their exact

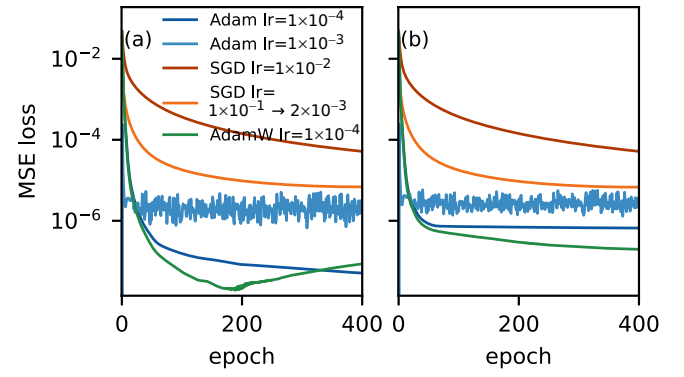


FIG. 12. Benchmarking of several stochastic optimizers on training topological model. Dataset is the same as in Fig. 2. The learning rates are indicated near each optimizer. For SGD optimizer, the variable learning rate with linear change from 10^{-1} to 2×10^{-3} was also tested.

expressions are

$$\text{MSE} = \frac{1}{\text{batch size}} \sum_{n=1}^{\text{batch size}} (W_n - W_{\text{prediction},n})^2, \quad (\text{E1})$$

for single-dimensional predictions of RSWN value W . For the case of one-hot encoding of four possible classes for extended

TABLE I. Models used in the study and their properties. The optimizers and learning rates are mentioned in the last column.

Model	Task	Structure	TN layer	Approximate number of trainable parameters for N -cell system	Optimizer	General note
“Topo”	Regression	4-leg tensors Dense layer	4-leg inputs	N^4	Adam, $lr = 10^{-4}$	Overparametrized for SSH
“Topo + sym”	Regression	2-leg tensors Dense layer	2-leg inputs	N^2	Adam, $lr = 10^{-4}$	Optimal for SSH
“Preproc + sym”	Regression	2-leg tensors Dense layer ReLU Dense layer Dense layer ReLU	2-leg inputs	N^2	Adam, $lr = 10^{-4}$ SGD, $lr = 10^{-2}$	Good optimization possibilities
“FNN”	Regression	Dense layer ReLU Dense layer	None	$4N^2$	Adam, $lr = 10^{-4}$ SGD, $lr = 10^{-2}$	Overparametrized for SSH
“Topo + sym”	Classification	2-leg tensors Dense layer	2-leg inputs	N^2	Adam, $lr = 10^{-4}$	optimal for SSH
“Preproc + sym”	Classification	2-leg tensors Dense layer ReLU Dense layer Dense layer ReLU	2-leg inputs	N^2	Adam, $lr = 10^{-4}$ SGD, $lr = 10^{-2}$	Good optimization possibilities
“FNN”	Classification	Dense layer ReLU Dense layer	None	$4N^2$	Adam, $lr = 10^{-4}$ SGD, $lr = 10^{-2}$	Overparametrized for SSH

SSH model, we use the cross-entropy formula:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \left(- \sum_{k=1}^4 y_{n,k} \ln \frac{\exp(z_{n,k})}{\sum_{j=1}^4 \exp(z_{n,j})} \right). \quad (\text{E2})$$

For each sample n , the neural network outputs a vector of logits $\mathbf{z}_n = (z_{n,1}, \dots, z_{n,4})$, one value per class. PyTorch internally applies a softmax to convert these logits into class probabilities. The target $\mathbf{y}_n$ is a one-hot vector, meaning that exactly one entry equals 1 (the correct class) and the others are 0 based on RSWN value. Because of this, the inner sum selects only the log probability of the

correct class. The loss is averaged over all N samples in the batch. ML model architectures are summarized in Table I.

To evaluate the performance of particular optimizers in training of topological neural network, we perform benchmarks on the same dataset described in the main text. The goal of the model is again to predict float value of RSWN. The results are shown in Fig. 12. The Adam optimizer with the learning rate 10^{-4} performs in the most stable way, while AdamW optimizer demonstrates even higher starting optimization speed, but then tends to worsen the loss eventually. SGD optimizers tested show lower accuracy and require much more epochs to reach similar MSE loss as Adam.

-
- [1] L. Wang, Discovering phase transitions with unsupervised learning, *Phys. Rev. B* **94**, 195105 (2016).
 - [2] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions and crossovers through unsupervised machine learning: A critical examination, *Phys. Rev. E* **95**, 062122 (2017).
 - [3] Z. Yue, Y. Wang, and P. Lyu, Incremental learning of phase transition in Ising model: Preprocessing, finite-size scaling and critical exponents, *Physica A* **600**, 127538 (2022).
 - [4] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, *Nat. Phys.* **13**, 431 (2017).
 - [5] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, *Phys. Rev. B* **96**, 195145 (2017).
 - [6] P. Zhang, H. Shen, and H. Zhai, Machine learning topological invariants with neural networks, *Phys. Rev. Lett.* **120**, 066401 (2018).
 - [7] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, *Nat. Phys.* **13**, 435 (2017).
 - [8] Y. Zhang, P. Ginsparg, and E.-A. Kim, Interpreting machine learning of topological quantum phase transitions, *Phys. Rev. Res.* **2**, 023283 (2020).
 - [9] I. Mondragon-Shem, T. L. Hughes, J. Song, and E. Prodan, Topological criticality in the chiral-symmetric AIII class at strong disorder, *Phys. Rev. Lett.* **113**, 046802 (2014).
 - [10] C.-K. Chiu, J. C. Y. Teo, A. P. Schnyder, and S. Ryu, Classification of topological quantum matter with symmetries, *Rev. Mod. Phys.* **88**, 035005 (2016).
 - [11] M. Z. Hasan and C. L. Kane, Colloquium: Topological insulators, *Rev. Mod. Phys.* **82**, 3045 (2010).
 - [12] X.-L. Qi and S.-C. Zhang, Topological insulators and superconductors, *Rev. Mod. Phys.* **83**, 1057 (2011).
 - [13] B. A. Bernevig, T. L. Hughes, and S.-C. Zhang, Quantum spin Hall effect and topological phase transition in HgTe quantum wells, *Science* **314**, 1757 (2006).
 - [14] L. Fu, Topological crystalline insulators, *Phys. Rev. Lett.* **106**, 106802 (2011).
 - [15] H. J. Zhang, C. X. Liu, X. L. Qi, X. Dai, Z. Fang, and S.-C. Zhang, Topological insulators in Bi₂Se₃, Bi₂Te₃ and Sb₂Te₃ with a single Dirac cone on the surface, *Nat. Phys.* **5**, 438 (2009).
 - [16] B. Yan and S.-C. Zhang, Topological materials, *Rep. Prog. Phys.* **75**, 096501 (2012).
 - [17] S. Rachel, Interacting topological insulators: A review, *Rep. Prog. Phys.* **81**, 116501 (2018).
 - [18] Y. Ando and L. Fu, Topological crystalline insulators and topological superconductors: From concepts to materials, *Annu. Rev. Condens. Matter Phys.* **6**, 361 (2015).
 - [19] S. de Léséleuc, V. Lienhard, P. Scholl, D. Barredo, S. Weber, N. Lang, H. P. Büchler, T. Lahaye, and A. Browaeys, Observation of a symmetry-protected topological phase of interacting bosons with Rydberg atoms, *Science* **365**, 775 (2019).
 - [20] L. J. Splitthoff, M. C. Belo, G. Jin, Y. Liu, E. Greplova, and C. K. Andersen, Gate-tunable phase transition in a resonator-based Su-Schrieffer-Heeger chain, *Phys. Rev. Res.* **6**, 043286 (2024).
 - [21] S. K. Kanungo, J. D. Whalen, Y. Lu, M. Yuan, S. Dasgupta, F. B. Dunning, K. R. A. Hazzard, and T. C. Killian, Realizing topological edge states with Rydberg-atom synthetic dimensions, *Nat. Commun.* **13**, 972 (2022).
 - [22] M. Kiczynski, S. Gorman, H. Geng, M. Donnelly, Y. Chung, Y. He, J. Keizer, and M. Simmons, Engineering topological states in atom-based semiconductor quantum dots, *Nature (London)* **606**, 694 (2022).
 - [23] V. Jouanny, S. Frasca, V. J. Weibel, L. Peyruchat, M. Scigliuzzo, F. Oppliger, F. De Palma, D. Sbroggio, G. Beaulieu, O. Zilberberg, *et al.*, Band engineering and study of disorder using topology in compact high kinetic inductance cavity arrays, *Nat. Commun.* **16**, 3396 (2025).
 - [24] F. Mei, G. Chen, L. Tian, S.-L. Zhu, and S. Jia, Robust quantum state transfer via topological edge states in superconducting qubit chains, *Phys. Rev. A* **98**, 012331 (2018).
 - [25] L.-N. Zheng, X. Yi, and H.-F. Wang, Engineering a phase-robust topological router in a dimerized superconducting-circuit lattice with long-range hopping and chiral symmetry, *Phys. Rev. Appl.* **18**, 054037 (2022).
 - [26] E. Kim, X. Zhang, V. S. Ferreira, J. Banker, J. K. Iverson, A. Sipahigil, M. Bello, A. González-Tudela, M. Mirhosseini, and O. Painter, Quantum electrodynamics in a topological waveguide, *Phys. Rev. X* **11**, 011015 (2021).
 - [27] C. Vega, M. Bello, D. Porras, and A. González-Tudela, Qubit-photon bound states in topological waveguides with long-range hoppings, *Phys. Rev. A* **104**, 053522 (2021).
 - [28] A. Altland and M. R. Zirnbauer, Nonstandard symmetry classes in mesoscopic normal-superconducting hybrid structures, *Phys. Rev. B* **55**, 1142 (1997).

- [29] A. P. Schnyder, S. Ryu, A. Furusaki, and A. W. W. Ludwig, Classification of topological insulators and superconductors in three spatial dimensions, *Phys. Rev. B* **78**, 195125 (2008).
- [30] A. Kitaev, Periodic table for topological insulators and superconductors, *AIP Conf. Proc.* **1134**, 22 (2009).
- [31] G. Jin, D. O. Oriekhov, L. J. Splitthoff, and E. Greplova, Topological finite size effect in one-dimensional chiral symmetric systems, [arXiv:2411.17822](https://arxiv.org/abs/2411.17822).
- [32] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber, Unsupervised identification of topological phase transitions using predictive models, *New J. Phys.* **22**, 045003 (2020).
- [33] N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai, Deep learning topological invariants of band insulators, *Phys. Rev. B* **98**, 085402 (2018).
- [34] Y. Zhang and E.-A. Kim, Quantum loop topography for machine learning, *Phys. Rev. Lett.* **118**, 216401 (2017).
- [35] P. Baireuther, M. Płodzień, T. Ojanen, J. Tworzydło, and T. Hyart, Identifying Chern numbers of superconductors from local measurements, *SciPost Phys. Core* **6**, 087 (2023).
- [36] M. D. Caio, M. Caccin, P. Baireuther, T. Hyart, and M. Fruchart, Machine learning assisted measurement of local topological invariants, [arXiv:1901.03346](https://arxiv.org/abs/1901.03346).
- [37] L.-F. Zhang, L.-Z. Tang, Z.-H. Huang, G.-Q. Zhang, W. Huang, and D.-W. Zhang, Machine learning topological invariants of non-Hermitian systems, *Phys. Rev. A* **103**, 012419 (2021).
- [38] P. Mognini, A. Zegarra, E. van Nieuwenburg, R. Chitra, and W. Chen, A supervised learning algorithm for interacting topological insulators based on local curvature, *SciPost Phys.* **11**, 073 (2021).
- [39] A. Ghosh and M. Sarkar, Supervised learning of an interacting two-dimensional hardcore boson model of a weak topological insulator using correlation functions, *Phys. Rev. B* **110**, 165134 (2024).
- [40] A. Ghosh, M. Sarkar, Y.-J. Kao, and P. Chen, Learning phases with quantum Monte Carlo simulation cell, *Mach. Learn.: Sci. Technol.* **6**, 045017 (2025).
- [41] W. P. Su, J. R. Schrieffer, and A. J. Heeger, Solitons in polyacetylene, *Phys. Rev. Lett.* **42**, 1698 (1979).
- [42] J. K. Asbóth, L. Oroszlány, and A. Pályi, *A Short Course on Topological Insulators: Band Structure and Edge States in One and Two Dimensions* (Springer, Cham, 2016).
- [43] B. Pérez-González, M. Bello, Á. Gómez-León, and G. Platero, SSH model with long-range hoppings: Topology, driving and disorder, [arXiv:1802.03973](https://arxiv.org/abs/1802.03973).
- [44] K. Cybinski, M. Płodzień, M. Tomza, M. Lewenstein, A. Dauphin, and A. Dawid, Characterizing out-of-distribution generalization of neural networks: Application to the disordered Su-Schrieffer-Heeger model, *Mach. Learn.: Sci. Technol.* **6**, 015014 (2025).
- [45] M. Tsang, D. Cheng, and Y. Liu, Detecting statistical interactions from neural network weights, in *Proceedings of the 6th International Conference on Learning Representations* (ICLR 2018).
- [46] A. Bibal and B. Frénay, in *ECAI 2020*, Frontiers in Artificial Intelligence and Applications, Vol. 325, pp. 1119–1126.
- [47] V. Hernandez, T. Spriggs, S. Khaleefah, and E. Greplová, Adiabatic fine-tuning of neural quantum states enables detection of phase transitions in weight space, [arXiv:2503.17140](https://arxiv.org/abs/2503.17140).
- [48] K. Cybinski, J. Enouen, A. Georges, and A. Dawid, Speak so a physicist can understand you! TetrisCNN for detecting phase transitions and order parameters, [arXiv:2411.02237](https://arxiv.org/abs/2411.02237).
- [49] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, KAN: Kolmogorov-Arnold networks, [arXiv:2404.19756](https://arxiv.org/abs/2404.19756).
- [50] D. Oriekhov, E. Greplova, B. Zougari, S. Bergkamp, G. Jin, N. El Yazidi, S. van der Meer, and J. D. Torres Luna, Code for paper: “Why is topology hard to learn?” [Computer software], Zenodo, 2025, <https://doi.org/10.5281/zenodo.17102795>; GitLab repository, <https://gitlab.com/QMAI/papers/toponeuralnetworks>.
- [51] S. Bergkamp, Topological phase transition learning. unsupervised and adversarial methods, master thesis, TU Delft, 2024.
- [52] B. Zougari, Simplifying neural networks for quantum classification, bachelor thesis, TU Delft, 2024.